ROMFØ.ASM

```
                        ;
                        ;       Code for ROM at F000 on TEI FDC-2
                        ;
F000                    org     0f000h
                        ;
                        ;       controller port equates
                        ;
0080 =          statreg equ     80h             ;status in
0080 =          cmdreg  equ     80h             ;commands out
0081 =          trkreg  equ     81h             ;track in/out
0082 =          sectreg equ     82h             ;sector in/out
0083 =          datareg equ     83h             ;data in/out
0084 =          drvsreg equ     84h             ;drive select
0084 =          hdldreg equ     84h             ;head load status
                        ;
                        ;       controller status equates
                        ;
0001 =          hdldst  equ     01h             ;head load bit
0010 =          skerst  equ     10h             ;seek error bit
0004 =          trk0st  equ     04h             ;track 0 bit
0001 =          busyst  equ     01h             ;busy bit
0002 =          drqst   equ     02h             ;data request bit
0080 =          nrst    equ     80h             ;not ready bit
                        ;
                        ;       controller error masks
                        ;
009C =          rderr   equ     10011100b       ;read
00FC =          wrerr   equ     11111100b       ;write
008C =          rterr   equ     10001100b       ;read track
00EC =          wterr   equ     11101100b       ;write track
0098 =          skerr   equ     10011000b       ;seek
000A =          maxtry  equ     10              ;max error retries
                        ;
                        ;       controller commands
                        ;
001D =          skcmd   equ     1dh             ;seek
001A =          skncmd  equ     1ah             ;seek no verify
0009 =          restcmd equ     09h             ;restore
0088 =          rdcmd   equ     88h             ;read sector
00A8 =          wrcmd   equ     0a8h            ;write sector
00E5 =          rtcmd   equ     0e5h            ;read track
00F4 =          wtcmd   equ     0f4h            ;write track
00D8 =          fintcmd equ     0d8h            ;interrupt command
                        ;
                        ;       interrupt address
                        ;
0010 =          intrupt equ     10h             ;interrupt address
0080 =          parms   equ     80h             ;boot buffer
                        ;
                        ;       transfer vector
                        ;
F000 C33EC0             jmp     0c03eh          ;to CUTER init
F003 C399F1             jmp     dskboot         ;general boot
```

```
F006 C330F0              jmp     dskinit         ;init controller
F009 C33FF0              jmp     dksel           ;drive select
F00C C351F0              jmp     dskhome         ;restore
F00F C362F0              jmp     dskseek         ;seek
F012 C3A3F0              jmp     dskrd           ;read
F015 C3D1F0              jmp     dskwr           ;write
F018 C3FFF0              jmp     dskrt           ;read track
F01B C32DF1              jmp     dskwt           ;write track
F01E C392F1              jmp     dskrtrk         ;read track #
F021 C395F1              jmp     dskwtrk         ;write track #
F024 C393F0              jmp     dsksnv          ;seek no verify
F027 C30000              jmp     0               ;for expansion
F02A C30000              jmp     0               ;for expansion
F02D C30000              jmp     0               ;for expansion
                ;
                ;       initialize disk controller
                ;
F030 =          dskinit  equ     $
F030 F3                  di                      ;no interrupts
F031 3EC9                mvi     a,0c9h          ;return
F033 321000              sta     intrupt
F036 FB                  ei                      ;interrupts ok now
F037 3ED8                mvi     a,fintcmd       ;reset controller
F039 D380                out     cmdreg
F03B CD5BF1              call    dskbusy         ;wait for controller
F03E C9                  ret
                ;
                ;       select a disk drive
                ;
                ;       c = drive #
                ;
                ;       0,1 8"; 2,3 5"
                ;
F03F =          dksel    equ     $
F03F E5                  push    h               ;save registers
F040 C5                  push    b
F041 214DF0              lxi     h,dsktab        ;select table
F044 0600                mvi     b,0             ;c=index
F046 09                  dad     b               ;index by drive #
F047 7E                  mov     a,m             ;get select command
F048 D384                out     drvsreg         ;select drive
F04A C1                  pop     b               ;restore registers
F04B E1                  pop     h
F04C C9                  ret
                ;
                ;       disk select table
                ;
F04D =          dsktab   equ     $
F04D 72                  db      72h             ;8" 1
F04E 74                  db      74h             ;8" 0
F04F B4                  db      10110100b       ;5" 0
F050 B2                  db      10110010b       ;5" 1
                ;
                ;       home disk to track 0
                ;
```

```
                ;       returns  z ok, nz error
                ;
F051 =          dskhome equ     $
F051 CD5BF1             call    dskbusy         ;wait for controller
F054 3E09               mvi     a,restcmd       ;restore command
F056 D380               out     cmdreg          ;do it
F058 CD5BF1             call    dskbusy         ;wait and get status
F05B E604               ani     trk0st          ;at home?
F05D CA51F0             jz      dskhome         ;no (????)
F060 AF                 xra     a               ;good status
F061 C9                 ret
                ;
                ;       seek routine
                ;
                ;       c = track #
                ;
                ;       returns  z ok, nz error
                ;
F062 =          dskseek equ     $
F062 C5                 push    b               ;save registers
F063 060A               mvi     b,maxtry        ;max retries
F065 CD5BF1             call    dskbusy         ;wait for controller
F068 DB81               in      trkreg          ;get current track
F06A B9                 cmp     c               ;the same?
F06B C275F0             jnz     dskseek1        ;nope
F06E DB84               in      hdldreg         ;is head loaded?
F070 E601               ani     hdldst
F072 CA90F0             jz      dskseek3        ;yes
F075 =          dskseek1 equ    $
F075 79                 mov     a,c             ;get desired track
F076 D383               out     datareg         ;tell controller for seek
F078 3E1D               mvi     a,skcmd         ;seek command
F07A D380               out     cmdreg          ;do it...
F07C CD5BF1             call    dskbusy         ;wait and get status
F07F E698               ani     skerr           ;errors?
F081 CA90F0             jz      dskseek3        ;no
F084 05                 dcr     b               ;count down retries
F085 3E01               mvi     a,1             ;pre set error status
F087 CA90F0             jz      dskseek3        ;out of tries
F08A CD51F0             call    dskhome         ;restore head
F08D CA75F0             jz      dskseek1        ;and try again
F090 =          dskseek3 equ    $
F090 C1                 pop     b               ;restore registers
F091 B7                 ora     a               ;set z/nz
F092 C9                 ret
                ;
                ;       seek without verify
                ;
F093 =          dsksnv  equ     $
F093 CD5BF1             call    dskbusy         ;wait for controller
F096 79                 mov     a,c             ;get track
F097 D383               out     datareg         ;tell controller
F099 3E1A               mvi     a,skncmd        ;seek no verify command
F09B D380               out     cmdreg          ;do it...
F09D CD5BF1             call    dskbusy         ;wait and get status
```

```
F0A0 E698               ani     skerr           ;check for errors
F0A2 C9                 ret                     ;return with status
                ;
                ;       read sector routine
                ;
                ;       c = sector
                ;       hl -> buffer
                ;
                ;       returns  z ok, nz error
                ;
F0A3 =          dskrd   equ     $
F0A3 C5                 push    b               ;save registers
F0A4 E5                 push    h               ;save buffer ptr
F0A5 060A               mvi     b,maxtry        ;max retries
F0A7 21BAF0             lxi     h,dskrdi        ;interrupt address
F0AA CD69F1             call    dsksetup        ;set up interrupt
F0AD =          dskrd1  equ     $
F0AD CD5BF1             call    dskbusy         ;wait for controller
F0B0 79                 mov     a,c             ;get sector
F0B1 D382               out     sectreg         ;tell controller
F0B3 E1                 pop     h               ;get a(buffer)
F0B4 E5                 push    h               ;save again
F0B5 3E88               mvi     a,rdcmd         ;read sector
F0B7 C37CF1             jmp     dskin           ;do transfer
F0BA =          dskrdi  equ     $               ;*** interrupt ***
F0BA 33                 inx     sp              ;discard return address
F0BB 33                 inx     sp
F0BC CD5BF1             call    dskbusy         ;wait and get status
F0BF E69C               ani     rderr           ;errors?
F0C1 CACAF0             jz      dskrd2          ;nope
F0C4 05                 dcr     b               ;count down retries
F0C5 C2ADF0             jnz     dskrd1          ;still some left
F0C8 3E01               mvi     a,1             ;set error status
F0CA =          dskrd2  equ     $
F0CA CD73F1             call    dskcln          ;clean up interrupt
F0CD E1                 pop     h               ;restore registers
F0CE C1                 pop     b
F0CF B7                 ora     a               ;set z/nz
F0D0 C9                 ret
                ;
                ;       write sector routine
                ;
                ;       c = sector
                ;       hl -> buffer
                ;
                ;       returns  z ok, nz error
                ;
F0D1 =          dskwr   equ     $
F0D1 C5                 push    b               ;save regs
F0D2 E5                 push    h               ;save buffer ptr
F0D3 060A               mvi     b,maxtry        ;max retries
F0D5 21E8F0             lxi     h,dskwri        ;interrupt address
F0D8 CD69F1             call    dsksetup        ;set up interrupt
F0DB =          dskwr1  equ     $
F0DB CD5BF1             call    dskbusy         ;wait for controller
```

```
F0DE 79              mov     a,c             ;get desired sector
F0DF D382            out     sectreg         ;tell controller
F0E1 E1              pop     h               ;get a(buffer)
F0E2 E5              push    h               ;save it again
F0E3 3EA8            mvi     a,wrcmd         ;write sector
F0E5 C387F1          jmp     dskout          ;do the transfer
F0E8 =       dskwri  equ     $               ;*** interrupt ***
F0E8 33              inx     sp              ;discard interrupt address
F0E9 33              inx     sp
F0EA CD5BF1          call    dskbusy         ;wait, get status
F0ED E6FC            ani     wrerr           ;error?
F0EF CAF8F0          jz      dskwr2          ;no
F0F2 05              dcr     b               ;decrement retries
F0F3 C2DBF0          jnz     dskwr1          ;try again
F0F6 3E01            mvi     a,1             ;set error status
F0F8 =       dskwr2  equ     $
F0F8 CD73F1          call    dskcln          ;clean up interrupt
F0FB E1              pop     h               ;restore registers
F0FC C1              pop     b
F0FD B7              ora     a               ;set z/nz
F0FE C9              ret
                     ;
                     ;       read track routine
                     ;
                     ;       c = sector to start on
                     ;       hl -> buffer
                     ;
                     ;       returns  z ok, nz error
                     ;
F0FF =       dskrt   equ     $
F0FF C5              push    b               ;save registers
F100 E5              push    h               ;save buffer ptr
F101 060A            mvi     b,maxtry        ;max retries
F103 2116F1          lxi     h,dskrti        ;interrupt address
F106 CD69F1          call    dsksetup        ;set up interrupt
F109 =       dskrt1  equ     $
F109 CD5BF1          call    dskbusy         ;wait for controller
F10C 79              mov     a,c             ;get sector
F10D D382            out     sectreg         ;tell controller
F10F E1              pop     h               ;get a(buffer)
F110 E5              push    h               ;save again
F111 3EE5            mvi     a,rtcmd         ;read track
F113 C37CF1          jmp     dskin           ;do transfer
F116 =       dskrti  equ     $               ;*** interrupt ***
F116 33              inx     sp              ;discard return address
F117 33              inx     sp
F118 CD5BF1          call    dskbusy         ;wait and get status
F11B E68C            ani     rterr           ;errors?
F11D CA26F1          jz      dskrt2          ;nope
F120 05              dcr     b               ;count down retries
F121 C209F1          jnz     dskrt1          ;still some left
F124 3E01            mvi     a,1             ;set error status
F126 =       dskrt2  equ     $
F126 CD73F1          call    dskcln          ;clean up interrupt
F129 E1              pop     h               ;restore registers
```

```
F12A C1                 pop     b
F12B B7                 ora     a               ;set z/nz
F12C C9                 ret
                ;
                ;       write track routine
                ;
                ;       c = sector to start on
                ;       hl -> buffer
                ;
                ;       returns  z ok, nz error
                ;
F12D =          dskwt   equ     $
F12D C5                 push    b               ;save regs
F12E E5                 push    h               ;save buffer ptr
F12F 060A               mvi     b,maxtry        ;max retries
F131 2144F1             lxi     h,dskwti        ;interrupt address
F134 CD69F1             call    dsksetup        ;set up interrupt
F137 =          dskwt1  equ     $
F137 CD5BF1             call    dskbusy         ;wait for controller
F13A 79                 mov     a,c             ;get desired sector
F13B D382               out     sectreg         ;tell controller
F13D E1                 pop     h               ;get a(buffer)
F13E E5                 push    h               ;save it again
F13F 3EF4               mvi     a,wtcmd         ;write track
F141 C387F1             jmp     dskout          ;do the transfer
F144 =          dskwti  equ     $               ;*** interrupt ***
F144 33                 inx     sp              ;discard interrupt address
F145 33                 inx     sp
F146 CD5BF1             call    dskbusy         ;wait, get status
F149 E6EC               ani     wterr           ;error?
F14B CA54F1             jz      dskwt2          ;no
F14E 05                 dcr     b               ;decrement retries
F14F C237F1             jnz     dskwt1          ;try again
F152 3E01               mvi     a,1             ;set error status
F154 =          dskwt2  equ     $
F154 CD73F1             call    dskcln          ;clean up interrupt
F157 E1                 pop     h               ;restore registers
F158 C1                 pop     b
F159 B7                 ora     a               ;set z/nz
F15A C9                 ret
                ;
                ;       wait for controller to be not busy
                ;       return with status
                ;
F15B =          dskbusy equ     $
F15B C5                 push    b               ;save registers
F15C =          dskbus1 equ     $
F15C DB80               in      statreg         ;read status
F15E 4F                 mov     c,a
F15F E601               ani     busyst          ;busy?
F161 C25CF1             jnz     dskbus1         ;wait...
F164 79                 mov     a,c             ;restore status
VF165 E6FE              ani     not(busyst)     ;kill busy bit
F167 C1                 pop     b               ;restore registers
F168 C9                 ret
```

```
                        ;
                        ;       set up i/o transfer interrupt routine
                        ;
F169 =          dsksetup equ    $
F169 F3                 di                      ;no interrupts
F16A 221100             shld    intrupt+1       ;branch address
F16D 3EC3               mvi     a,0c3h          ;jump
F16F 321000             sta     intrupt
F172 C9                 ret
                        ;
                        ;       i/o transfer interrupt routine clean up
                        ;
F173 =          dskcln  equ     $
F173 F5                 push    psw             ;save a
F174 3EC9               mvi     a,0c9h          ;return
F176 321000             sta     intrupt         ;simple interrupt handler
F179 F1                 pop     psw             ;restore a
F17A FB                 ei                      ;interrupts ok now
F17B C9                 ret
                        ;
                        ;       input transfer routine
                        ;
                        ;       a = command to use
                        ;       hl -> buffer
                        ;
                        ;       returns via interrupt
                        ;
F17C =          dskin   equ     $
F17C FB                 ei                      ;interrupts on
F17D D380               out     cmdreg          ;start transfer
F17F =          dskin1  equ     $
F17F 76                 hlt                     ;wait for disk
F180 DB83               in      datareg         ;get data
F182 77                 mov     m,a             ;store into buffer
F183 23                 inx     h               ;bump buffer ptr
F184 C37FF1             jmp     dskin1          ;and loop
                        ;
                        ;       output transfer loop
                        ;
                        ;       a = command to use
                        ;       hl -> buffer
                        ;
                        ;       returns via interrupt
                        ;
F187 =          dskout  equ     $
F187 FB                 ei                      ;interrupts ok
F188 D380               out     cmdreg          ;start transfer
F18A =          dskout1 equ     $
F18A 76                 hlt                     ;wait for disk...
F18B 7E                 mov     a,m             ;get data
F18C D383               out     datareg         ;send to disk
F18E 23                 inx     h               ;bump buffer ptr
F18F C38AF1             jmp     dskout1         ;and loop
                        ;
                        ;       return current controller track
```

```
                ;
F192 =          dskrtrk equ     $
F192 DB81               in      trkreg          ;read controller track
F194 C9                 ret
                ;
                ;       set current controller track
                ;
F195 =          dskwtrk equ     $
F195 79                 mov     a,c             ;get track #
F196 D381               out     trkreg          ;tell the controller
F198 C9                 ret
                ;
                ;       general bootstrap
                ;
                ;       sector 1 has boot data
                ;
                ;               a(to load)
                ;               length to load in sectors
                ;               a(entry)
                ;
                ;
F199 =          dskboot equ     $
F199 310002             lxi     sp,200h         ;set up a stack
F19C CD30F0             call    dskinit         ;init controller
F19F 0E00               mvi     c,0             ;drive 0
F1A1 CD3FF0             call    dsksel
F1A4 CD51F0             call    dskhome         ;restore
F1A7 C299F1             jnz     dskboot         ;keep trying
F1AA 0E00               mvi     c,0             ;seek, load head
F1AC CD62F0             call    dskseek
F1AF C299F1             jnz     dskboot         ;keep trying
F1B2 0E01               mvi     c,1             ;sector 1
F1B4 218000             lxi     h,parms         ;parms buffer
F1B7 CDA3F0             call    dskrd           ;read parms
F1BA C2BAF1             jnz     $               ;read error
F1BD 2A8200             lhld    parms+2         ;length to load
F1C0 EB                 xchg                    ;de = sector count
F1C1 2A8000             lhld    parms           ;load address
F1C4 010200             lxi     b,2             ;b=track, c=sector
F1C7 =          dskboot1 equ    $
F1C7 CDA3F0             call    dskrd           ;read it in
F1CA C2CAF1             jnz     $               ;read error
F1CD 1D                 dcr     e               ;any left to do
F1CE CAEDF1             jz      dskboot2        ;no
F1D1 D5                 push    d               ;save counter
F1D2 118000             lxi     d,128           ;sector length
F1D5 19                 dad     d               ;bump buffer pointer
F1D6 D1                 pop     d               ;restore counter
F1D7 0C                 inr     c               ;next sector
F1D8 79                 mov     a,c             ;sector in a
F1D9 FE1B               cpi     27              ;too big?
F1DB C2C7F1             jnz     dskboot1        ;no, go read
F1DE 0E01               mvi     c,1             ;reset sector to 1
F1E0 04                 inr     b               ;increment track
F1E1 C5                 push    b               ;save track and sector
```

```
F1E2 48              mov     c,b             ;track in c
F1E3 CD62F0          call    dskseek         ;step track
F1E6 C2E6F1          jnz     $               ;seek error
F1E9 C1              pop     b               ;restore sector and track
F1EA C3C7F1          jmp     dskboot1        ;and go read
F1ED =         dskboot2 equ  $
F1ED 2A8400          lhld    parms+4         ;transfer address
F1F0 E9              pchl                    ;off we go.....
F1F1                 end
```